Lecture #8 CS 695 1.2.2023

Quiz #1 — 3rd Feb in-class

[mem mgmt]

[live migration] [post copy]

[remus] [insight]

[snowflock]

(*) IO virtualization for VMs

~ system-view of ~~machine~~ OS includes devices.

~ device (external devices)

CPU ⟺ (interconnect) ⟺ devic.

CPU { interface } device

(*) device drivers use the interface to communicate w/ devices.

- OS uses the device drivers to translate actions from its own subsystems.

atleast 3 ways for the OS to communicate ...

(i) PIO — programmed IO.

↳ special ISA instructions to deal w/ IO.

↳ eg. in, out ——— serial port.

↳ port index, data

(ii) MMIO ↳ memory mapped IO

↳ phy-mem is mapped to device interface state.

↳ r/w mem ⟺ r/w to device regs.

(iii) DMA :~ where devices r/w directly to/from memory.

↳ setup by instruction on CPU by the OS.

e.g: consider a disk, with following interface

reg x : sector number
y : R/w
z : VA (memory address)
in/out : instr. to state purpose

---

x ⇐ 10
y ⇐ 0
z ⇐ kva
in

read from sector 10 & copy to kva (*) specified in reg. z.

vd sector | psector
10 → 1346
0

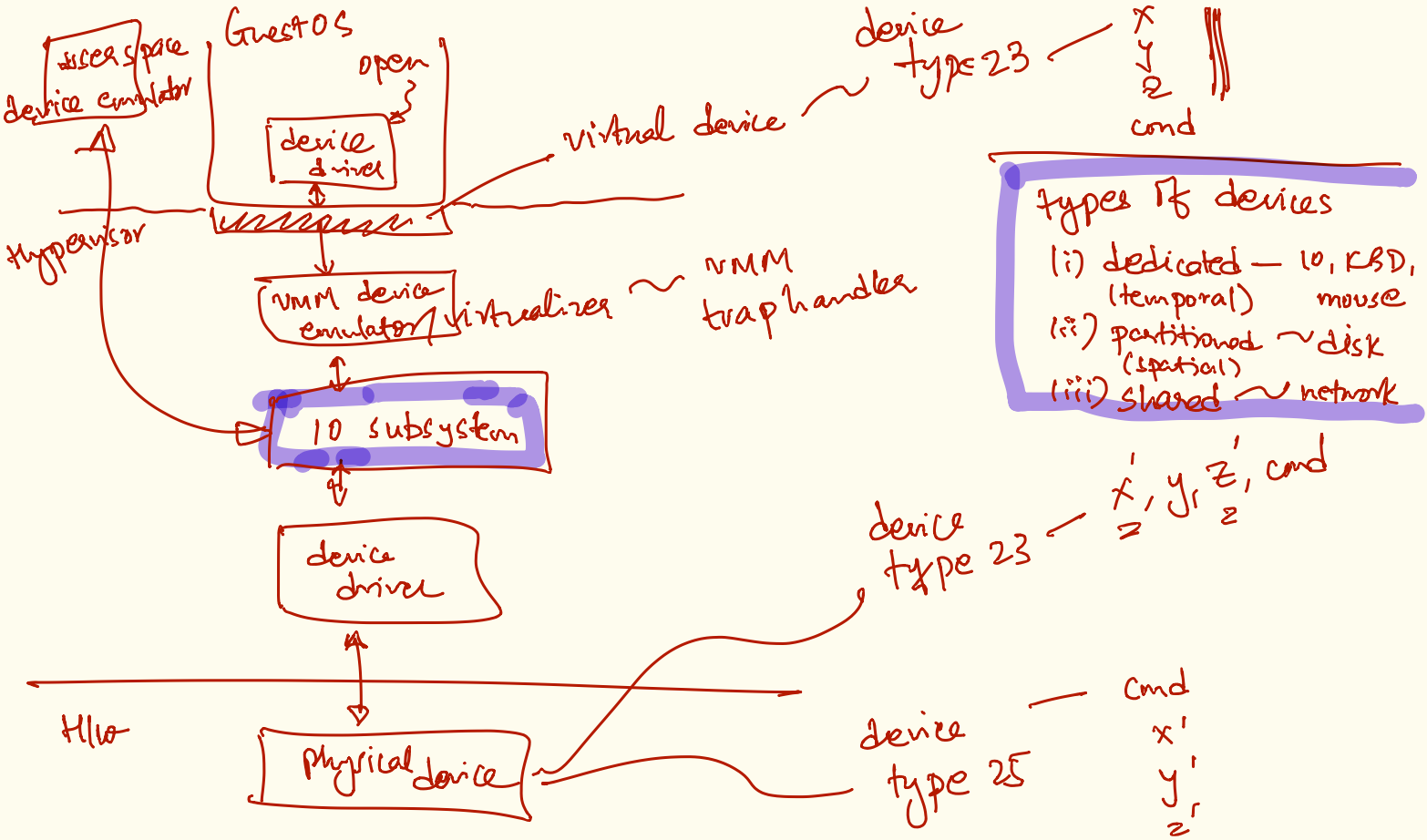
(*) kva hyp.

privileged instn.

fopen → open fd → device driver (device block num) → x, y, z interface

---

(#) Generalized IO virtualization view with VMs.



(*) virtual device may/may not be the same as the physical device.

(*) virtual device maybe virtualized via diff. abstractions. e.g: vdisk ↔ file

① Full device virtualization (fully virtualized VMs).

~ guest OS does not know about the VMM

~ VMM virtualizes and handles/emulates the complete device interface.

~

* actual eg: e1000 NIC

* 2 complete IO stacks:

device type 23

x
y
z
cmd. ← (i) privileged instr. generates trap.

(ii) patch instr. to gen. trap.

(iii) write-protect mmio pages to generate trap.

② PV IO virtualization — split device (driver) model.

device type 23

x ← IO
y ← 0
z ← Kva
cmdregs ← 0/1

trap
trap
trap
trap

PV

single hypercall

hc-deviceIO (x, y, z, cmd, devid)

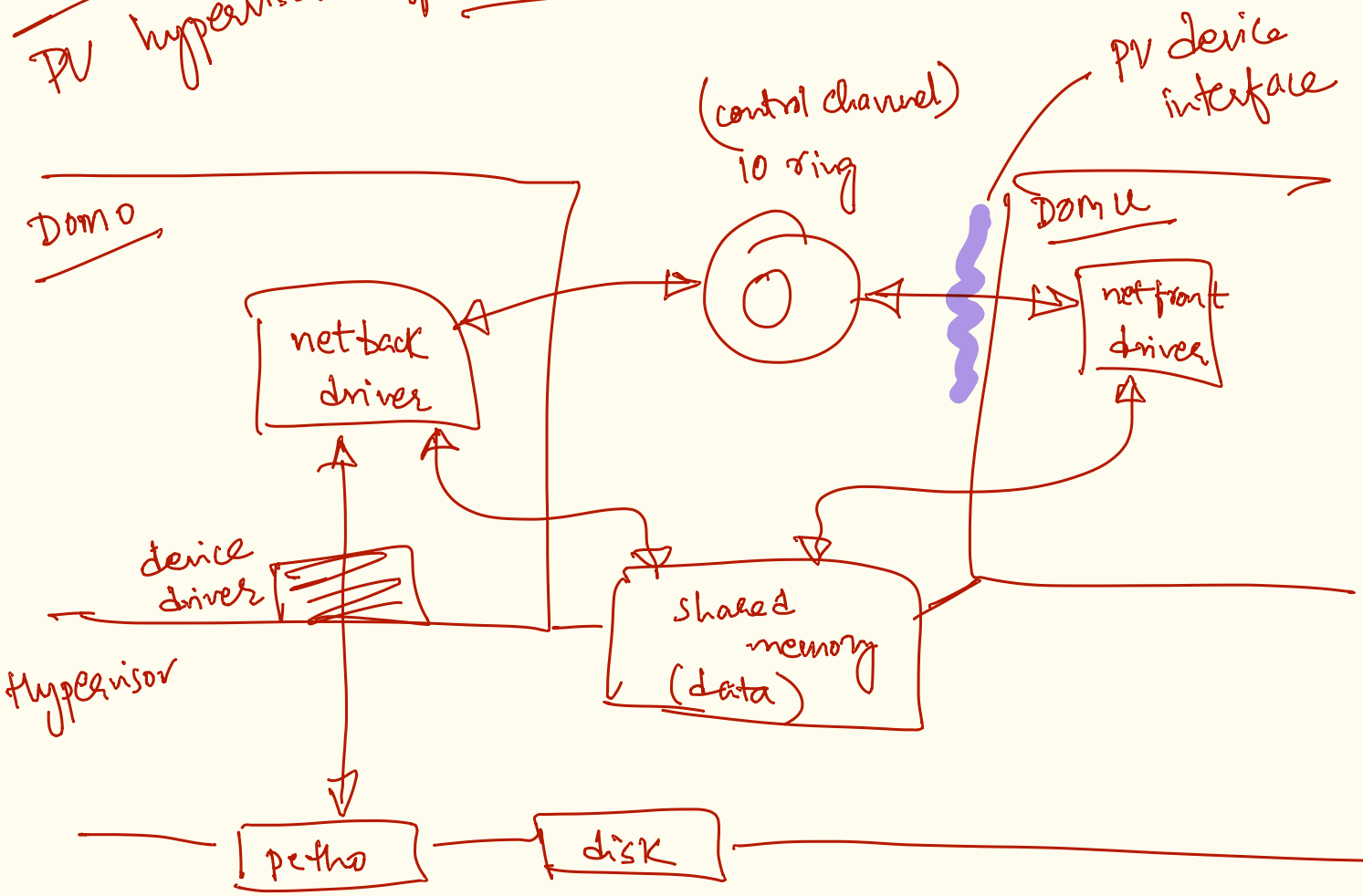
* change device interface to a virtual interface
non-physical/non-real

⇒ send all control+data info in a single hypercall.

⇒ pack multiple units of IO in a single call.

→ simplify device specification for greater efficiency!

Xen
PV hypervisor || virtio (KVM, vmware) ..



Q1. How deep should the n/w stack be on the guest OS?

Q2. How does Dom0 get/receive pkts destined for DomU?